IN THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A computer implemented method for dynamically rendering data in a markup language, the method comprising:

identifying a symbol in the data in the markup language, the symbol indicating a query of a data set, the query containing one or more variables, each variable of one of a plurality of data types[[;]], wherein

~~augmenting~~ the markup language is augmented with a variable resolution functionality to support the variables ~~by building a variable resolution functionality into the markup language~~, each variable resolving to two or more variable values;

accessing the data set in order to generate a resolution to the query, wherein the one or more variables contained in the query are resolved as part of the generation of the resolution to the query, the query associated with a tag in the markup language;

substituting the two or more variable values for each variable into the query to generate two or more completed queries; and

dynamically rendering the resolution to the two or more completed queries together as a part of the markup language, according to at least one rule associated with the markup language wherein said symbol can be used to dynamically render multiple data sets.

2. (Original) The method of claim 1, wherein:

the symbol comprises a delimited token.

3. (Original) The method of claim 1 wherein:

the symbol is located within the data in the markup language such that the query is associated with a markup language tag.

4. (Original) The method of claim 3 wherein:

the markup language comprises Hyper Text Markup Language.

5. (Previously Presented) The method of claim 3 wherein rendering further comprises:

rendering the resolution according to at least one rule associated with the markup language tag with which the query is associated.

6. (Original) The method of claim 1 wherein:

the data set comprises a set of at least one document in a hierarchically structured format.

7. (Original) The method of claim 6 wherein:

the hierarchically structured format comprises Extensible Markup Language.

8. (Original) The method of claim 7 wherein:

the symbol conforms an Extensible Markup Language standard concerning queries.

9. (Original) The method of claim 1 wherein:

the data set comprises a database.

10. (Canceled)

11. (Original) The method of claim 1 wherein:

rendering is performed by software running on a hand held computing device.

12. (Original) The method of claim 1 further comprising:

generating a resolution to the query by retrieving a node set from a set of documents in Extensible Markup Language; and

rendering each member of the node set.

13. (Canceled)

14. (Previously presented) The method of claim 1 wherein:

each variable contained in the query comprises a token bounded by delimiters.

15-16. (Canceled)

17. (Previously presented) The method of claim 1 wherein rendering the resolution further comprises:

receiving an input from a user;

responsive to receiving the input, updating the data set based at least in part on the received input.

18. (Canceled)

19. (Currently Amended) A computer program product for dynamically rendering data in a markup language, the computer program product comprising:

program code for identifying a symbol in the data in the markup language, the symbol indicating a query of a data set, the query containing one or more variables, each variable of one of a plurality of data types;

program code for augmenting the markup language to support the variables by building a variable resolution functionality into the markup language, each variable resolving to two or more variable values;

program code for accessing the data set in order to generate a resolution to the query, wherein the one or more variables contained in the query are resolved as part of the generation of the resolution to the query, the query associated with a tag in the markup language;

program code for substituting the two or more variable values for each variable into the query to generate two or more completed queries;

program code for dynamically rendering the resolution to the two or more completed queries together as a part of the markup language, according to at least one rule associated with the markup language wherein said symbol can be used to dynamically render multiple data sets; and

a computer readable medium on which the program codes are stored.

20. (Previously Presented) The computer program product of claim 19 further comprising:

program code for rendering the resolution according to at least one rule associated with a markup language tag with which the query is associated.

21. (Original) The computer program product of claim 19 further comprising:

program code for generating a resolution to the query by retrieving a node set from a set of documents in Extensible Markup Language; and

program code for rendering each member of the node set.

22. (Previously presented) The computer program product of claim 19 further comprising:

program code for:

receiving an input from a user;

responsive to receiving the input, updating the data set based at least in part on the received input.

23. (Canceled)

24. (Currently Amended) A computer system for dynamically rendering data in a markup language, the computer system comprising:

a computer-readable storage medium storing an executable computer program product comprising:

an identification module, for identifying a symbol in the data in the markup language, the symbol indicating a query of a data set, the query containing one or more variables, each variable of one of a plurality of data types, ~~the identification module further adapted to augment~~ the markup language augmented with a variable resolution functionality to support the variables ~~by building a variable resolution functionality into the markup language~~, each variable resolving to two or more variable values;

a data access module, for accessing the data set in order to generate a resolution to the query, wherein the one or more variables contained in the query are resolved as part of the generation of the resolution to the query, the query associated with a tag in the markup language, the data access module being coupled to the identification module, the data access module further adapted to substitute the two or more variable values for each variable into the query to generate two or more completed queries; and

a rendering module, for dynamically rendering the resolution to the two or more completed queries together as a part of the markup language, according to at least one rule associated with the markup language wherein said symbol can be used to dynamically render multiple data sets, the rendering module being coupled to the data access module.

25. (Previously Presented) The system of claim 24 wherein:

the rendering module is further for rendering the resolution according to at least one rule associated with a markup language tag with which the query is associated.

26. (Original) The system of claim 24 further comprising:

a resolution generation module, for generating a resolution to the query by retrieving a node set from a set of documents in Extensible Markup Language, the resolution generation module being coupled to the data access module; and

the rendering module is further for rendering each member of the node set.

27. (Previously presented) The system of claim 24 further comprising:

an updating module, for receiving an input from a user and responsive to receiving the input updating the data set based at least in part on the received input, the updating module being coupled to the rendering module.

28. (Canceled)

29. (Original) The method of claim 3 wherein:

the markup language comprises Wireless Markup Language.

30. (Currently Amended) The method of claim 1 wherein augmenting the markup language with a variable resolution functionality to support the variables further comprises:

providing a variable table for storing names and values of the variables, each variable of one of the plurality of data types; and

utilizing a syntax in the markup language for creating the variables by adding the variables to the variable table.

31. (Previously presented) The method of claim 1, wherein each completed query comprises a node and the at least one rule is associated with a tag describing cells of a rendered table and directs rendering of each node belonging to each completed query as a cell of the rendered table.

32. (Previously presented) The computer program product of claim 19, wherein each completed query comprises a node and the at least one rule is associated with a tag describing cells of a rendered table and directs rendering of each node belonging to each completed query as a cell of the rendered table.

33. (Previously presented) The computer system of claim 24, wherein each completed query comprises a node and the at least one rule is associated with a tag describing cells of a rendered table and directs rendering of each node belonging to each completed query as a cell of the rendered table.